Laporan Tugas Besar
Mata Kuliah Teori Bahasa dan Automata
“Pengolahan Bahasa Jerman (Germany)”

Laporan ini dibuat dengan tujuan untuk memenuhi tugas mata kuliah
Teori Bahasa dan Automata

Dosen Pengampu :

Disusun oleh :
1. Maulana Malik Ibrahim - 1301204222
2. Kelvyn Lukito - 1301200104
3. Satrio Tejo Kusumo - 1301204484

Laporan Tugas Besar

Program Studi Informatika
Fakultas Informatika
Universitas Telkom
2022

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa. Atas rahmat dan hidayah-Nya, kami bisa menyelesaikan laporan yang berjudul "*Laporan Tugas Besar Mata Kuliah Teori Bahasa dan Automata "Pengolahan Bahasa Jerman (Germany)"*". Tidak lupa kami mengucapkan terima kasih kepada Bapak Ahmad Suryan selaku guru Mata Kuliah Teori Bahasa dan Automata. Laporan ini memberikan panduan dalam pengolahan Bahasa Jerman. Bagi pembaca untuk memahami *Lexical Analyzer* pengolahan Bahasa Jerman.

Daftar Isi

Program Utama

Beberapa file yang digunakan untuk menjalankan program utama lexical analyzer dan parser.

- Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./style.css">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.cs
s" rel="stylesheet"
integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWu
tUpFmBp4vmVor" crossorigin="anonymous">
    <!--<script type="text/javascript" src="./lexicalAnalyzer.js"></script>-->
    <title>Tugas Besar TBA</title>
</head>
<body>
    <div class="card card-lexical main-card" style="width: 40rem;">
        <div class="card-body">
          <h5 class="card-title">lexical Analyzer & Parser</h5>
          <p class="card-text">This Section is the lexical Analyzer. Fill the form with 3
words according to SVO rules</p>
          <form>
            <input id="inputValue" type="text" placeholder="Subject Verb Object" >
            <button id="button-primary" type="button" onclick="lexicalAnalyzer()">
              Submit
            </button>
          </form>
          <div class="alert alert-primary" role="alert" style="margin-top: 10px;">
            <p id="token1">
              Token 1
            </p>
            <p id="token2">
              Token 2
            </p>
            <p id="token3">
              Token 3
            </p>
            <p id="result-lexical">
              Lexical Will be Here
            </p>
            <p id="result-parser">
              Parser Will be Here
```

```
          </p>
        </div>
      </div>
    </div>
    <script src="./lexicalAnalyzer.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2" crossorigin="anonymous"></script>

</body>
</html>
```

- style.css

```
*{
    font-family: poppins;
}
.card {
    margin: auto;
    margin-top: 50px;
    margin-bottom: 50px;
}
.card-parser h5{
    font-weight: bold;
}

.card-lexical h5{
    font-weight: bold;
}

#button-primary{
    background-color: rgb(55, 101, 250);
    padding: 12px 20px 12px 20px;
    color: white;
    font-family: poppins;
    border-radius: 5px;
    border-width: none;
    width: 38rem;
}

#inputValue{
    border-radius: 5px;
    padding: 10px 20px 10px 20px;
    border-width: 2px;
    border-color: grey;
    width: 38rem;
    margin-bottom: 12px;
}
```

- lexicalAnalyzer.js

```
// Initialization
const alphabet =
["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w","x","y","z"];

state_list =["q1","q2","q3","q4","q5","q6","q7","q8","q9","q10","q11","q12","q13","q14","q15",
"q16","q17","q18","q19","q20","q21","q22","q23","q24","q25","q26","q27","q28","q29","q30",
"q31",
"q32","q33","q34","q35","q36","q37","q38","q39","q40","q41","q41","q43","q44","q45","q46",
"q47","q48",
"q49","q50","q51","q52","q53","q54","q55","q56","q57","q58","q59","q60","q61","q62","q63",
"q64","q65",
"q66","q67","q68","q69","q70","q71","q72","q73","q74","q75","q76","q78","q79","q80","q81",
"q82","q83",
"q84","q85","q86","q87","q88","q89","q90","q91","q92","q93","q93","q94","q95","q96","q97",
"q98","q99",
"q100"];

/*for (let i=0;i<100;i++){
    state_list[i] = `$q{i+1}`;
}
*/

transition_list = {}

for (var state, _pj_c = 0, _pj_a = state_list, _pj_b = _pj_a.length; _pj_c < _pj_b; _pj_c += 1)
{
    state = _pj_a[_pj_c];

    for (var alpa, _pj_f = 0, _pj_d = alphabet, _pj_e = _pj_d.length; _pj_f < _pj_e; _pj_f +=
1) {
      alpa = _pj_d[_pj_f];
      transition_list[[state, alpa]] = "error";
    }

    transition_list[[state, "#"]] = "error";
    transition_list[[state, " "]] = "error";
  }

// Symbol Definition
/*
  S = Terminal
  SA = SUBJECT
  VB = VERB
  OB = OBJECT
*/
var non_terminals,terminals,parse_tabel
non_terminals = ["S","SA","VB","OB"]
terminals = ["vater","mutter","bruder","er","ich","sie","onkel","tante","wir","du",
```

```
"lessen","essen","sehen","benutzen","offen","ritten","finden","putzen","waschen","bekomen
",

"physik","fleisch","konzert","kleid","schuh","waschen","wagen","Lebensmittel","flugzeug","g
eld"]

// Parse Tabel Definition
parse_tabel = {}

parse_tabel[["S", "vater"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "mutter"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "bruder"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "er"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "ich"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "sie"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "onkel"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "tante"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "wir"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "du"]] = ["SA", "VB", "OB"];

parse_tabel[["S", "lessen"]] = ["error"];
parse_tabel[["S", "essen"]] = ["error"];
parse_tabel[["S", "sehen"]] = ["error"];
parse_tabel[["S", "benutzen"]] = ["error"];
parse_tabel[["S", "offen"]] = ["error"];
parse_tabel[["S", "ritten"]] = ["error"];
parse_tabel[["S", "finden"]] = ["error"];
parse_tabel[["S", "putzen"]] = ["error"];
parse_tabel[["S", "waschen"]] = ["error"];
parse_tabel[["S", "bekomen"]] = ["error"];

parse_tabel[["S", "physik"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "fleisch"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "konzert"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "kleid"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "schuh"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "waschen"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "wagen"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "lebensmittel"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "flugzeug"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "geld"]] = ["SA", "VB", "OB"];
parse_tabel[["S", "EOS"]] = ["error"];

parse_tabel[["SA", "vater"]] = ["vater"];
parse_tabel[["SA", "mutter"]] = ["mutter"];
parse_tabel[["SA", "bruder"]] = ["bruder"];
parse_tabel[["SA", "er"]] = ["er"];
parse_tabel[["SA", "ich"]] = ["ich"];
parse_tabel[["SA", "sie"]] = ["sie"];
parse_tabel[["SA", "onkel"]] = ["onkel"];
parse_tabel[["SA", "tante"]] = ["tante"];
parse_tabel[["SA", "wir"]] = ["wir"];
parse_tabel[["SA", "du"]] = ["du"];
```

```
parse_tabel[["SA", "lessen"]] = ["error"];
parse_tabel[["SA", "essen"]] = ["error"];
parse_tabel[["SA", "sehen"]] = ["error"];
parse_tabel[["SA", "benutzen"]] = ["error"];
parse_tabel[["SA", "offen"]] = ["error"];
parse_tabel[["SA", "ritten"]] = ["error"];
parse_tabel[["SA", "finden"]] = ["error"];
parse_tabel[["SA", "putzen"]] = ["error"];
parse_tabel[["SA", "waschen"]] = ["error"];
parse_tabel[["SA", "bekomen"]] = ["error"];

parse_tabel[["SA", "physik"]] = ["error"];
parse_tabel[["SA", "fleisch"]] = ["error"];
parse_tabel[["SA", "konzert"]] = ["error"];
parse_tabel[["SA", "kleid"]] = ["error"];
parse_tabel[["SA", "schuh"]] = ["error"];
parse_tabel[["SA", "waschen"]] = ["error"];
parse_tabel[["SA", "wagen"]] = ["error"];
parse_tabel[["SA", "lebensmittel"]] = ["error"];
parse_tabel[["SA", "flugzeug"]] = ["error"];
parse_tabel[["SA", "geld"]] = ["error"];
parse_tabel[["SA", "EOS"]] = ["error"];

parse_tabel[["VB", "vater"]] = ["error"];
parse_tabel[["VB", "mutter"]] = ["error"];
parse_tabel[["VB", "bruder"]] = ["error"];
parse_tabel[["VB", "er"]] = ["error"];
parse_tabel[["VB", "ich"]] = ["error"];
parse_tabel[["VB", "sie"]] = ["error"];
parse_tabel[["VB", "onkel"]] = ["error"];
parse_tabel[["VB", "tante"]] = ["error"];
parse_tabel[["VB", "wir"]] = ["error"];
parse_tabel[["VB", "du"]] = ["error"];

parse_tabel[["VB", "lessen"]] = ["lessen"];
parse_tabel[["VB", "essen"]] = ["essen"];
parse_tabel[["VB", "sehen"]] = ["sehen"];
parse_tabel[["VB", "benutzen"]] = ["benutzen"];
parse_tabel[["VB", "offen"]] = ["offen"];
parse_tabel[["VB", "ritten"]] = ["ritten"];
parse_tabel[["VB", "finden"]] = ["finden"];
parse_tabel[["VB", "putzen"]] = ["putzen"];
parse_tabel[["VB", "waschen"]] = ["waschen"];
parse_tabel[["VB", "bekomen"]] = ["bekomen"];

parse_tabel[["VB", "physik"]] = ["error"];
parse_tabel[["VB", "fleisch"]] = ["error"];
parse_tabel[["VB", "konzert"]] = ["error"];
parse_tabel[["VB", "kleid"]] = ["error"];
parse_tabel[["VB", "schuh"]] = ["error"];
parse_tabel[["VB", "waschen"]] = ["error"];
parse_tabel[["VB", "wagen"]] = ["error"];
parse_tabel[["VB", "lebensmittel"]] = ["error"];
parse_tabel[["VB", "flugzeug"]] = ["error"];
```

```
parse_tabel[["VB", "geld"]] = ["error"];
parse_tabel[["VB", "EOS"]] = ["error"];

parse_tabel[["OB", "vater"]] = ["error"];
parse_tabel[["OB", "mutter"]] = ["error"];
parse_tabel[["OB", "bruder"]] = ["error"];
parse_tabel[["OB", "er"]] = ["error"];
parse_tabel[["OB", "ich"]] = ["error"];
parse_tabel[["OB", "sie"]] = ["error"];
parse_tabel[["OB", "onkel"]] = ["error"];
parse_tabel[["OB", "tante"]] = ["error"];
parse_tabel[["OB", "wir"]] = ["error"];
parse_tabel[["OB", "du"]] = ["error"];

parse_tabel[["OB", "lessen"]] = ["error"];
parse_tabel[["OB", "essen"]] = ["error"];
parse_tabel[["OB", "sehen"]] = ["error"];
parse_tabel[["OB", "benutzen"]] = ["error"];
parse_tabel[["OB", "offen"]] = ["error"];
parse_tabel[["OB", "ritten"]] = ["error"];
parse_tabel[["OB", "finden"]] = ["error"];
parse_tabel[["OB", "putzen"]] = ["error"];
parse_tabel[["OB", "waschen"]] = ["error"];
parse_tabel[["OB", "bekomen"]] = ["error"];

parse_tabel[["OB", "physik"]] = ["physik"];
parse_tabel[["OB", "fleisch"]] = ["fleisch"];
parse_tabel[["OB", "konzert"]] = ["konzert"];
parse_tabel[["OB", "kleid"]] = ["kleid"];
parse_tabel[["OB", "schuh"]] = ["schuh"];
parse_tabel[["OB", "waschen"]] = ["waschen"];
parse_tabel[["OB", "wagen"]] = ["wagen"];
parse_tabel[["OB", "lebensmittel"]] = ["lebensmittel"];
parse_tabel[["OB", "flugzeug"]] = ["flugzeug"];
parse_tabel[["OB", "geld"]] = ["geld"];
parse_tabel[["OB", "EOS"]] = ["error"];


// space before input string
transition_list[["q1", " "]] = "q1";

// subjek mutter
transition_list[["q1", "m"]] = "q2";
transition_list[["q2", "u"]] = "q3";
transition_list[["q3", "t"]] = "q4";
transition_list[["q4", "t"]] = "q5";
transition_list[["q5", "e"]] = "q6";
transition_list[["q6", "r"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";

// subjek vater
```

```
transition_list[["q1", "v"]] = "q8";
transition_list[["q8", "a"]] = "q4";
transition_list[["q4", "t"]] = "q5";
transition_list[["q5", "e"]] = "q6";
transition_list[["q6", "r"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";

// subjek bruter
transition_list[["q1", "b"]] = "q14";
transition_list[["q14", "r"]] = "q15";
transition_list[["q15", "u"]] = "q16";
transition_list[["q16", "d"]] = "q5";
transition_list[["q5", "e"]] = "q6";
transition_list[["q6", "r"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";


// subjek onkel
transition_list[["q1", "o"]] = "q21";
transition_list[["q21", "n"]] = "q22";
transition_list[["q22", "k"]] = "q23";
transition_list[["q23", "e"]] = "q24";
transition_list[["q24", "l"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";


// subjek tante
transition_list[["q1", "t"]] = "q11";
transition_list[["q11", "a"]] = "q12";
transition_list[["q12", "n"]] = "q13";
transition_list[["q13", "t"]] = "q10";
transition_list[["q10", "e"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";


// subjek ich
transition_list[["q1", "i"]] = "q19";
transition_list[["q19", "c"]] = "q20";
transition_list[["q20", "h"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
```

```
transition_list[["q25", " "]] = "q25";

// subjek sie
transition_list[["q1", "s"]] = "q9";
transition_list[["q9", "i"]] = "q10";
transition_list[["q10", "e"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";

// subjek wir
transition_list[["q1", "w"]] = "q17";
transition_list[["q17", "i"]] = "q6";
transition_list[["q6", "r"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";

// subjek du
transition_list[["q1", "d"]] = "q18";
transition_list[["q18", "u"]] = "q7";
transition_list[["q7", "#"]] = "ACCEPT";
transition_list[["q7", ""]] = "ACCEPT";
transition_list[["q7", " "]] = "q25";
transition_list[["q25", " "]] = "q25";

// verb  lessen
transition_list[["q25", "l"]] = "q26";
transition_list[["q26", "e"]] = "q27";
transition_list[["q27", "s"]] = "q28";
transition_list[["q28", "s"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb essen
transition_list[["q25", "e"]] = "q27";
transition_list[["q27", "s"]] = "q28";
transition_list[["q28", "s"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb waschen
transition_list[["q25", "w"]] = "q30";
transition_list[["q30", "a"]] = "q31";
```

```
transition_list[["q31", "s"]] = "q32";
transition_list[["q32", "c"]] = "q33";
transition_list[["q33", "h"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb sehen
transition_list[["q25", "s"]] = "q34";
transition_list[["q34", "e"]] = "q33";
transition_list[["q33", "h"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb offen
transition_list[["q25", "o"]] = "q35";
transition_list[["q35", "f"]] = "q36";
transition_list[["q36", "f"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb bekomen

transition_list[["q38", "k"]] = "q39";
transition_list[["q39", "o"]] = "q40";
transition_list[["q40", "m"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb benutzen
transition_list[["q25", "b"]] = "q37";
transition_list[["q37", "e"]] = "q38";
transition_list[["q38", "n"]] = "q41";
transition_list[["q41", "u"]] = "q42";
transition_list[["q42", "t"]] = "q43";
transition_list[["q43", "z"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
```

```
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb putzen
transition_list[["q25", "p"]] = "q41";
transition_list[["q41", "u"]] = "q42";
transition_list[["q42", "t"]] = "q43";
transition_list[["q43", "z"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb ritten
transition_list[["q25", "r"]] = "q44";
transition_list[["q44", "i"]] = "q45";
transition_list[["q45", "t"]] = "q46";
transition_list[["q46", "t"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";

// verb finden
transition_list[["q25", "f"]] = "q47";
transition_list[["q47", "i"]] = "q48";
transition_list[["q48", "n"]] = "q49";
transition_list[["q49", "d"]] = "q29";
transition_list[["q29", "e"]] = "q50";
transition_list[["q50", "n"]] = "q51";
transition_list[["q51", "#"]] = "ACCEPT";
transition_list[["q51", ""]] = "ACCEPT";
transition_list[["q51", " "]] = "q52";
transition_list[["q52", " "]] = "q52";


// objek  physik
transition_list[["q52", "p"]] = "q53";
transition_list[["q53", "h"]] = "q54";
transition_list[["q54", "y"]] = "q55";
transition_list[["q55", "s"]] = "q56";
transition_list[["q56", "i"]] = "q57";
transition_list[["q57", "k"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek fleisch
transition_list[["q52", "f"]] = "q59";
transition_list[["q59", "l"]] = "q60";
transition_list[["q60", "e"]] = "q66";
```

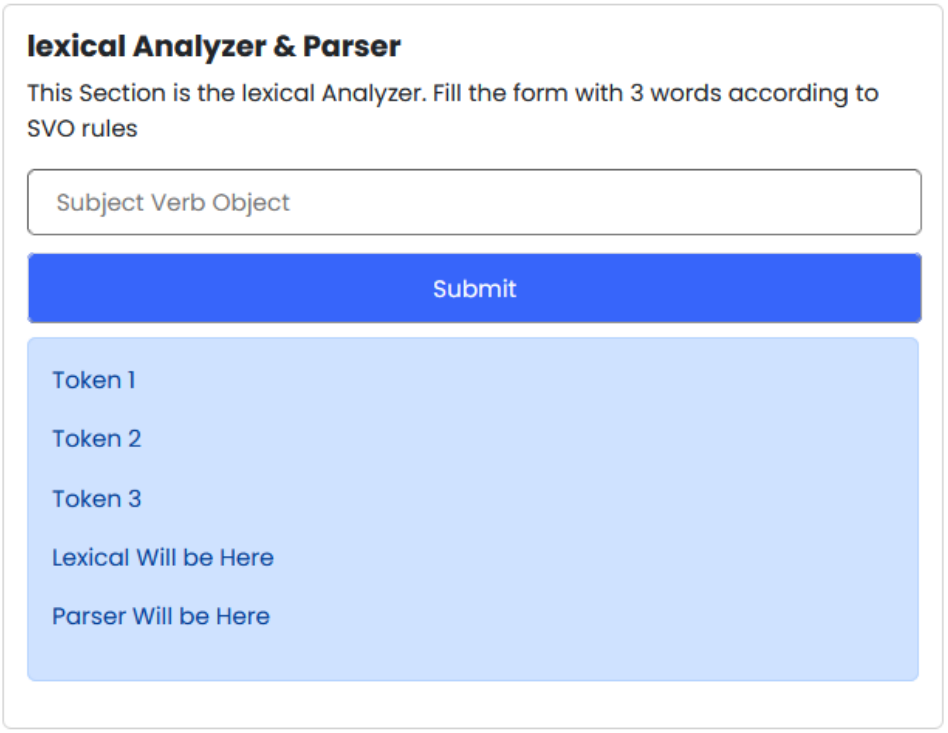
```
transition_list[["q66", "i"]] = "q67";
transition_list[["q67", "s"]] = "q68";
transition_list[["q68", "c"]] = "q69";
transition_list[["q69", "h"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek konzert
transition_list[["q52", "k"]] = "q70";
transition_list[["q70", "o"]] = "q71";
transition_list[["q71", "n"]] = "q72";
transition_list[["q72", "z"]] = "q73";
transition_list[["q73", "e"]] = "q74";
transition_list[["q74", "r"]] = "q75";
transition_list[["q75", "t"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek kleid
transition_list[["q52", "k"]] = "q70";
transition_list[["q70", "l"]] = "q76";
transition_list[["q76", "e"]] = "q77";
transition_list[["q77", "i"]] = "q78";
transition_list[["q78", "d"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek schuh
transition_list[["q52", "s"]] = "q81";
transition_list[["q81", "c"]] = "q82";
transition_list[["q82", "h"]] = "q83";
transition_list[["q83", "u"]] = "q69";
transition_list[["q69", "h"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek waschen

transition_list[["q52", "w"]] = "q95";
transition_list[["q95", "a"]] = "q96";
transition_list[["q96", "s"]] = "q97";
transition_list[["q97", "c"]] = "q98";
transition_list[["q98", "h"]] = "q99";
transition_list[["q99", "e"]] = "q100";
transition_list[["q100", "n"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek wagen
transition_list[["q52", "w"]] = "q95";
transition_list[["q95", "a"]] = "q96";
transition_list[["q96", "g"]] = "q99";
transition_list[["q99", "e"]] = "q100";
transition_list[["q100", "n"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek lebensmittel
transition_list[["q52", "l"]] = "q84";
transition_list[["q84", "e"]] = "q85";
```

```
transition_list[["q85", "b"]] = "q86";
transition_list[["q86", "e"]] = "q87";
transition_list[["q87", "n"]] = "q88";
transition_list[["q88", "s"]] = "q89";
transition_list[["q89", "m"]] = "q90";
transition_list[["q90", "i"]] = "q91";
transition_list[["q91", "t"]] = "q92";
transition_list[["q92", "t"]] = "q93";
transition_list[["q93", "e"]] = "q94";
transition_list[["q94", "l"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek flugzeug
transition_list[["q52", "f"]] = "q59";
transition_list[["q59", "l"]] = "q60";
transition_list[["q60", "u"]] = "q61";
transition_list[["q61", "g"]] = "q62";
transition_list[["q62", "z"]] = "q63";
transition_list[["q63", "e"]] = "q64";
transition_list[["q64", "u"]] = "q65";
transition_list[["q56", "g"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";

// objek geld
transition_list[["q52", "g"]] = "q79";
transition_list[["q79", "e"]] = "q80";
transition_list[["q80", "l"]] = "q78";
transition_list[["q78", "d"]] = "q58";
transition_list[["q58", "#"]] = "ACCEPT";


const lexicalAnalyzer = () =>{
    // Input Sentence User
    var inputValue = document.querySelector("#inputValue").value;
    text = inputValue.toLowerCase()+"#";
    var token1,token2,token3;
    var result_lexical;
    var idx_char = 0;
    var state = "q0";
    var current_token = '';
    while (state !== "ACCEPT"){
        var current_char = text[idx_char];
        current_token += current_char;
        state = transition_list[(state, current_char)];
        if (state === "q25"){
            token1 = current_token;
        }
        if (state === "q52"){
            token2 = current_token;
        }
        if (state === "q58"){
            token3 = current_token
        }
        if (state === "error"){
```

```
        result_lexical = "error";
        break;
    }
    idx_char = idx_char + 1;
}
if (state === "ACCEPT"){
    result_lexical = "Valid"
}

// Initilization stack for parser
var parse_stack
parse_stack =[]
parse_stack.push("#")
parse_stack.push("S")
// parsing process
var idx_token,symbol
idx_token = 0
symbol = tokens[idx_token]

while (length(parse_stack) > 0){
    var top = parse_stack[length(parse_stack)-1];
    console.log('top= ',top);
    console.log('symbol= ',symbol);
    for (let i=0;i<length(terminals);i++){
        if (terminals[i]==top){
            console.log("Top adalah simbol terminal");
            if (top===symbol){
                parse_stack.pop();
                idx_token += 1;
                symbol = tokens[idx_token];
                if (symbol==="EOS"){
                    console.log("isi stack :",parse_stack);
                    parse_stack.pop();
                }
            } else {
                console.log("error");
                break;
            }
        }
    }
    for (let j=0;j<length(non_terminals);j++){
        if (top===non_terminals[j]){
            console.log("Top adalah simbol terminal");
            if (parse_tabel[[top,symbol]][0]!=="error"){
                parse_stack.pop();
                var symbol_to_be_pushed = parse_tabel[[top,symbol]];
                for (var i = symbol_to_be_pushed.length - 1, _pj_a = -1; i < _pj_a; i += -1) {
                    parse_stack.push(symbol_to_be_pushed[i]);
                }
            } else {
                console.log("error");
                break;
            }
```

```
            } else {
                console.log("error");
                break;
            }
            console.log("Isi Stack : ", parse_stack);
            console.log();
        }
    }
    // conslusion
    console.log();
    if (symbol==="EOS" && length(parse_stack)===0){
        console.log(`Input Value: ${inputValue} diterima, Aturan sesuai grammar`);
        document.querySelector("#result-parser").innerHTML =`Input Value: ${inputValue}
diterima, Aturan sesuai grammar`;
    } else {
        document.querySelector("#result-parser").innerHTML =`Input Value: ${inputValue}
tidak diterima, Aturan tidak sesuai grammar`;
    }

    document.querySelector("#token1").innerHTML = `Current Token: ${token1} valid`;
    //print("current token: ",current_token)
    document.querySelector("#token2").innerHTML = `Current Token: ${token2} valid`;
    //print("current token: ",current_token)
    document.querySelector("#token3").innerHTML = `Current Token: ${token3} valid`;
    //print("current token: ",current_token)
    if (result_lexical==="error"){
        document.querySelector("#result-lexical").innerHTML = result_lexical;
        //print("error")
    } else {
        document.querySelector("#result-lexical").innerHTML = `Semua Token di Input:
${inputValue} ${result_lexical}`;
        //print("Semua Token di Input: ",inputValue,"valid")
    }

  }
```

Tampilan Website



Tata Cara Penggunaan

1. Download Semua File dari repository github : https://github.com/kelvynlukito/LexicalAnalyzer-Parser
2. Jalankan file **index.html**
3. Masukkan input lexical analyzer pada form yang tersedia dan input akan diolah pada file **lexicalAnalyzer.js**
4. Rule of Input
   S → **<subjek> <verb> <objek>**
   subjek → **vater | mutter | bruder | er | ich | sie | onkel | tante | wir | du**
   verb → **lessen | essen | sehen | benutzen | offen | ritten | finden | putzen | waschen | bekomen**
   objek → **physik | fleisch | konzert | kleid | schuh | waschen | wagen | Lebensmittel | flugzeug | geld**
5. Example of Input : onkel offen wagen
6. klik submit dan hasil program akan tercetak.
7. Note : Jika hasil output program tidak keluar gunakan link dibawah sebagai cadangan : https://colab.research.google.com/drive/13wUsxwAyQmRce_yE2zXJ3lPByeqpwhYS#scrollTo=dySjQKbAdcC5